

2-1 値と変数

値には「型」がある

まず理解すべきは、プログラミングの基礎となる値についてです。値にはいくつかの種類、単純なデータやデータをまとめた構造などがあります。この種類を型といいます。

Pythonではtype関数で値の型の情報を取得できます。

```
type(1) # int
```

基本となる値の型を整理しましょう。

●——数値 - Numeric Types(int, float, complex)

「数字」のデータ型は数値(Numeric Types)と総称されます。Pythonでは、数字の値は、そのまま普通に数値を書くだけです。特別なことは何もありません。

数値は、更に「整数(int)」と「浮動小数点数(float)」「複素数(complex)」に分けて考えられます。この中でよく使うのは整数と浮動小数点数でしょう。整数と比較してより精度が求められる数値が必要なときは、浮動小数点数を使います。整数と浮動小数点数の例を示します。

・整数の例

```
2
```

```
200
```

```
-2
```

・浮動小数点数の例、ドットを含む数字と覚える

```
1.1
```

```
3.0
```

```
2.100002
```

4-1 正規表現

文字列処理を効率化する正規表現

本章では、文書进行处理することについて学んでいきます。まず文字列データを処理するときに必須となる正規表現を身につけましょう。

文字列処理を考えると、外すことのできないものが**正規表現**です。正規表現は、文字列をある決まったルールによるパターンとして定義し、それをもとに検索や置換を行うものです。パターンを使うため、例えば「数字だけを取り出す」とか「メールアドレスだけを取り出す」というようなことも可能になります。反面、「パターンをどう定義するか」によって取り出す文字列が大きく変わるため、慣れないうちは使いこなすのが難しい技術でもあります。

正規表現は、さまざまなプログラミング言語でサポートされています。Pythonの場合は、`re` という標準ライブラリとして用意されています。

```
import re
```

文字列を置換する

文字列の置換を行うために用意されているのが「`sub`」関数です。まずはこれを使って正規表現がどういったものか体験しましょう。この関数は、3つの必須の引数を持ちます。第1引数が、正規表現のパターン(検索文字列)です。第2引数に置換する文字列、第3引数に処理の対象となる文字列を指定します。[]で囲んでいる `flags` はオプションな引数で、検索方法の指定などに用います。

```
re.sub(パターン, 置換文字列, 処理する文字列 [, flags=フラグ ] )
```

●——正規表現による置換の違い

正規表現がどんなものか知るために、例として3章で作成した文字列置換の処理(「3-3 文字列処理」のリスト3-8)を正規表現を用いた形で書き直してみます。`little`を`BIG`に置換し、結果を表示します。`little`は、大文字小文字が違っていてもすべて置換されます。

5-3 JSON/XMLの活用

JSONを使う

Webサイトで公開されているデータには、HTML以外にもさまざまな形式のものがあります。中でも特に多用されているのが、「JSON」でしょう。

JSONは「JavaScript Object Notation」の略で、軽量のデータ交換用フォーマットとして用いられているものです。名前の通りJavaScriptのオブジェクト記法を参考に作成された記法です。構造的なデータをWebで配布するために広く使われます。JSONのデータの扱い方を知っておくことは、Webスクレイピングを行う上で非常に重要です。

PythonでJSONを扱うには、「json」という標準ライブラリを利用します。

```
import json
```

JSONを利用するのに必要な機能は、実は2つだけです。「JSONからPythonで扱いやすいデータに変換する機能」と「PythonのデータをJSON形式に変換する機能」です。Pythonのオブジェクトに変換できれば、後は好きなように処理できるでしょう。

・JSONをPythonオブジェクトに変換する

```
json.load( ファイル )  
json.loads( テキスト )
```

JSONのファイルやテキストを読み込み、Pythonのオブジェクトに変換するものです。loadでは、ファイル(fileオブジェクト)を引数に指定すると、そのファイルのデータを読み込みPythonオブジェクトに変換して返します。loadsは、引数にJSONデータの文字列を指定すると、それをPythonオブジェクトに変換して返します。

どちらの関数も、この他に各種の設定を行うための引数を持っていますが、ファイルや文字列を引数に指定して実行する方法さえ知っていれば基本的な変換は行えるでしょう。

・PythonオブジェクトをJSONに変換する

```
json.dump( 値 )  
json.dumps( 値 )
```

```
<title>{{title}}</title>
<style>
p { font-size: 20pt; }
tr th { padding:5px; background-color:black; color:white; }
tr td { padding:5px; border:1px solid gray;}
</style>
</head>
<body>
<h1>{{title}}</h1>
<table>
  <tr>
    <th>ID</th>
    <th>NAME</th>
    <th>MAIL</th>
  </tr>
  {% for item in data %}
  <tr>
    <td>{{item.id}}</td>
    <td>{{item.name}}</td>
    <td>{{item.mail}}</td>
  </tr>
  {% endfor %}
</table>
</body>
</html>
```

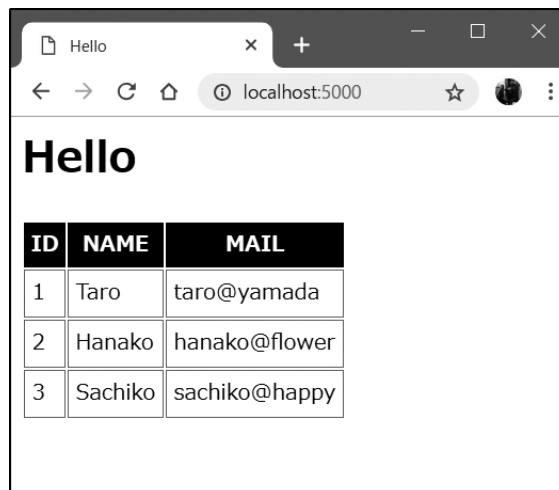


図 6-9 : data の内容をテーブルにまとめて表示したところ。

Web ブラウザで表示すると、data にまとめられたデータがテーブルの形で表示されるのが確認

7-3 TensorFlowでディープラーニング

TensorFlowを準備する

機械学習ライブラリは、scikit-learn以外にもさまざまなものがあります。中でも、もっとも注目度が高いのは「TensorFlow^{*5}」でしょう。Googleによって開発されたオープンソースの機械学習ライブラリです。TensorFlowはディープラーニング(深層学習)と呼ばれる技術を利用できるライブラリです。

●—— TensorFlowのインストール

Navigatorを起動し、左側のリストから「Environment」をクリックして使用している仮想環境を選択します。そして上部の「Installed」と表示されたボタンをクリックし「All」を選択します。その右側の検索フィールドに「tensorflow」と入力し検索します。いくつかのものが検索されますが、「tensorflow」という名前のをチェックして「Apply」ボタンでインストールします。標準Pythonの環境なら `pip install tensorflow` でインストールします。

Column

Anacondaは正式サポートされていない?

AnacondaでTensorFlowを利用する場合、留意しておきたいのは、「TensorFlowが正式サポートしている環境ではない」という点です。TensorFlowはpipのみを正式サポートしており、Anaconda(のパッケージ管理システムであるconda)は正式サポートしていません。

ただし、TensorFlow側から見れば非公式ですが、Anacondaのパッケージ管理システムにもTensorFlowは用意されており、pipを利用した場合とほぼ同様に利用することができます。またAnacondaの開発元によれば、conda版はpip版よりも最大8倍高速に実行される^{*a}とのことで、Anaconda版を利用する利点も大いにあります。

少なくとも学習目的であれば、Anacondaでも全く問題ないと考えていいでしょう。

* a <https://www.anaconda.com/blog/developer-blog/tensorflow-in-anaconda/>

* 5 <https://www.tensorflow.org/>